Rule-based Formalization of Eligibility Criteria for Clinical Trials

Huang, Z.; ten Teije, A.C.M.; van Harmelen, F.A.H.

published in
Lecture Notes in Computer Science
2013

DOI (link to publisher)
10.1007/978-3-642-38326-7_7

document version
Early version, also known as pre-print

Rule-based Formalization of Eligibility Criteria for Clinical Trials

Zhisheng Huang, Annette den Teije, and Frank van Harmelen

Department of Computer Science,
VU University Amsterdam, The Netherlands
{huang,annette,Frank.van.Harmelen}@cs.vu.nl

Abstract. In this paper, we propose a rule-based formalization of eligibility criteria for clinical trials. The rule-based formalization is implemented by using the logic programming language Prolog. Compared with existing formalizations such as pattern-based and script-based languages, the rule-based formalization has the advantages of being declarative, expressive, reusable and easy to maintain. Our rule-based formalisation is based on a general framework for eligibility criteria containting three types of knowledge: (1) trial-specific knowledge, (2) domain-specific knowledge and (3) common knowledge. This framework enables the reuse of several parts of the formalisation of eligibility criteria. We have implemented the proposed rule-based formalization in SemanticCT, a semantically-enabled system for clinical trials, showing the feasibility of using our rule-based formalization of eligibility criteria for supporting patient recruitment in clinical trial systems.

1 Introduction

Clinical trials have played important roles in medical research and drug development, because they provide sets of tests which generate safety and efficacy data for health interventions. However, the work in clinical trials have been considered to be laborious, sometimes, exhausting, because many procedures in clinical trials, such as patient recruitment (i.e., finding eligible patients for a trial) and trial finding (i.e., finding suitable trials for a patient), usually require manually processes. The goal of the formalizations of eligibility criteria is to provide faster identification of patients for trials and automatic identification of clinical trials for patients. That requires the implementation of the advanced reasoning services for matching patient data with formalized eligibility criteria.

There have been several attempts to the formulizations of eligibility criteria for clinical trials, which include pattern-based formalization, semantic-annotation–based formalization, and script-based formalization [1]. Compared with existing formalizations, a rule-based formalization is expected to be efficient and effective, because of their declarative nature, their high expressiveness, their reusability and easy maintenance. We have implemented the proposed rule-based formulization in SemanticCT, a semantically-enabled system for clinical trials [2][1]. SemanticCT provides semantic integration of various data in clinical trials. The system is designed to be a semantically-enabled system for decision support in various settings of clinical trials. In this paper we will show that the rule-based formalization of eligibility criteria is an efficient approach to identifying eligible patients for clinical trials.

This paper is organized as follows: After a brief motivation for rules as a formalisation language and for Prolog as a corresponding implementation language (Section 2), we present a small framework that captures three different types of knowledge that play a role in eligibility criteria (Section 3). Section 4 then discusses the formalization of these different knowledge in terms of

[1] http://wasp.cs.vu.nl/sct

rules. Section 5 discusses our implementation of this formalisation, and presents some experiments that show the feasibility of our approach. Section 6 discusses related work, concludes, and discusses the future work.

2 Motivation for Rules and Prolog

2.1 Rules

A rule-based formalization is expected to be an efficient and effective formalism to support automatic patient recruitment and trial feasibility testing, because of the following features:

Declarative. A rule-based formalization is a declarative language that expresses the logic of a computation without the need of exactly describing its control flow. That is significantly different from traditional programming languages, like Java, which use a procedural approach for the specification of control flow in the computation. A declarative formalization is more suitable for knowledge representation and reasoning because it needs no carefully design its computation (or reasoning) procedure. Thus, a rule-based formalization of eligibility criteria would provide a more convenient and efficient way for the automatic patient recruitments in clinical trials, compared with other procedural approaches, like the script-based formalization, which relies procedural scripts, and the pattern-based approach, which is based on SPARQL queries with regular expressions.

Easy Maintenance. Rule-based formalization provides an approach in which specified knowledge is easy to be understood for human users, because they are very close to human knowledge. It would not be too hard for human users to check the correctness of the specification of eligibility criteria if they are formalized as a set of rules. Furthermore, changing or revising a single rule would not make an effect on other part of the formalization significantly, because the meaning of the specification is usually represented in the specific rule. Thus, it is much easier for maintenance of knowledge, compared with procedural/scripting approaches of the formalization of eligibility criteria.

Reusability. In a rule-based formalization, a single rule (or a set of rules) is usually considered to be independent from other part of knowledge. Thus, it is much more convenient to re-use some rules of a formalized clinical trial for formalization of other trials. Furthermore, some rules which specify common knowledge, such as rules for temporal reasoning, and domain knowledge, and those that specify knowledge of diseases, can be designed to be a common library, which can be re-used for the formalization of other trials.

Expressiveness. Automatic patient recruitment usually involves comprehensive scenarios of deliberation and decision-making procedures. To facilitate those capabilities, it may require sophisticated data processing in workflows. An expressive rule-based language can support various functionalities of data processing. Thus, it provides the possibility to build workflows for various scenarios of medical applications.

2.2 Rule-based Language Prolog

There exist various rule languages which can be used for the formalization of eligibility criteria. In the researches of artificial intelligence, logic programming languages, like Prolog, are well known and popular rule-based languages. Several rule-based languages, like SWRL[2] and RIF[3], have been proposed for the semantics-enable rule-based language. In biomedical domain, the Arden syntax [4] has been developed to formalize rule-like medical knowledge. However, compared with

[2] http://www.w3.org/Submission/SWRL/
[3] http://www.w3.org/TR/rif-overview/
[4] http://www.hl7.org/special/Committees/arden/index.cfm

logic programming language Prolog, both SWRL, RIF and the Arden syntax have very limited functionalities for data processing.

In this paper, we will propose a rule-based formalization, which is based on the logic programming language Prolog. Prolog is a general purpose logic programming language associated with artificial intelligence, in particular, for knowledge representation and reasoning.

In Prolog, program logic is expressed in terms of relations. More exactly, those relations are formalized as a set of the predicates, like those in the first order logic. A computation is initiated by running a query over these relations.

In Prolog, the relations are represented as an atom which consists of a predicate with several terms as its parameters, like this: age_between(PatientData, AgeMin, AgeMax).

A rule in Prolog has the following form

```
Head :- Body.
```

Where Head is an atomic formula, and Body is a list of atomic formulas which are separated with commas and ends with a dot. For example,

```
triple_negative(Patient):-
   er_negative(Patient), pr_negative(Patient), her2_negative(Patient).
```

Which means that a patient of breast cancer is triple negative if it is ER negative, PR negative, and HER2 negative.

More exactly, our rule-based formalization is developed based on the SWI-Prolog[5]. The reasons why we select SWI-Prolog as the basic language for the rule-based formalization, because of the following features of it's powerful libraries and its semantic web support SWI-Prolog [3].

3 Framework

In this section we will first sketch the structure of typical eligibility criteria, and based on this we will describe a simpleframework that captures three different types of knowledge that typically occur in such eligibility criteria

3.1 Eligibility Criteria

Eligibility criteria consist of inclusion criteria, which state a set of conditions that must be met, and exclusion criteria, which state a set of conditions that must not be met, in order to participate in a clinical trial.

Take the example of the trial NCT00002720, the eligibility criteria are:

```
DISEASE CHARACTERISTICS:
  -  Histologically proven stage I, invasive breast cancer
  -  Hormone receptor status:
     -  Estrogen receptor positive
     -  Progesterone receptor positive or negative
PATIENT CHARACTERISTICS:
  Age: 65 to 80,
  Sex: Female
  Menopausal status: Postmenopausal
  Other: - No serious disease that would preclude surgery
         - No other prior or concurrent malignancy except basal cell
           carcinoma or carcinoma in situ of the cervix
```

[5] http://www.swi-prolog.org/

Those inclusion criteria (such as 'invasive breast cancer') and exclusion criteria (such as 'No serious disease that would preclude surgery') are trial specific. However, in order to check whether or not a required item (i.e., a criterion) has been met by a patient record, we need some domain knowledge to interpret the requirement and make it directly checkable from patient data. For example, 'invasive breast cancer' can be defined as either 'invasive ductal carcinoma' or 'invasive lobular carcinoma' in the diagnosis. Furthermore, we need some knowledge, such as temporal reasoning knowledge, to deal with temporal aspects of criteria, and service interface knowledge, to get the corresponding patient data from the EHR or CMR servers.

3.2 Different Knowledge Levels

We can formalize the knowledge rules of the specification of eligibility criteria of clinical trials with respect to the following different re-usable knowledge types:

Trial-specific Knowledge. Trial-specific knowledge are those rules which specify the concrete details of the eligibility criteria of a specific clinical trial. Those criteria are different from one trial to another. This is the formalisation of which specific inclusion criteria and exclusion criteria are required for a particular clinical trial. The formalisation of the criteria themselves are part of the other levels of knowledge.

Domain-specific Knowledge. Those trial-specific rules above may involve some knowledge which is domain specific, but that domain knowledge is in principle trial independent. Sch domain specific, but trial independent knowledge can be formalised in re-usable libraries. For example, for clinical trials of breast cancer, we formalize the knowledge of breast cancer in the knowledge bases of breast cancer, a domain-specific library of rules. An example ot this type of knowledge is a patient of breast cancer is triple nagative if the patient has estrogon receptor negative, progesterone receptor negative and protein HER2 negative status.

Common Knowledge. The specification of the eligibility criteria may involve some knowledge which is domain independent, like for example knowledge about temporal reasoning and the knowledge for manipulating semantic data and interacting with data servers, e.g. how to obtain the data from SPARQL endpoints. We formalize those knowledge in several rule libraries, which can again be reused across different applications.

4 Formalization

4.1 Formalization of Trial-specific knowledge

For the specification of eligibility criteria, we usually formalize their inclusion criteria and exclusion criteria respectively.

Given a patient ID, we suppose that we can obtain its patient data through the common knowledge of the interface with SPARQL endpoints and its internal data storage. Thus, in order to check if a patient meets an inclusion criterion, we can check if its patient data meet the criterion. Furthermore, we would not expect to check all the criteria with respect to the patient data, because some of those required data may be missing in the patient data. We introduce a special predicate getNotYetCheckedItems to collect those criteria which have not yet been formalized for the trial.

The inclusion criteria in the trial NCT00002720 above can be formalized in the following:

```
meetInclusionCriteria(_PatientID, PatientData, CT, NotYetCheckedItems):-
                      CT = 'nct00002720',
                      breast_cancer_stage(PatientData, '1'),
                      invasive_breast_cancer(PatientData),
                      er_positive(PatientData),
```

```
                    known_pr_status(PatientData),
                    age_between(PatientData, 65, 80),
                    postmenopausal(PatientData),
                    getNotYetCheckedItems(CT, NotYetCheckedItems).
```

There are no exclusion criteria for the trial 'NCT00002720'[6]. Thus, we formalize it as follow:

```
meetExclusionCriteria(_PatientID, _PatientData, CT):- CT = 'nct00002720', false.
```

If there is an exclusion criteria, like this: 'no currently pregnant', we can formalize the exclusion criteria as follows:

```
meetExclusionCriteria(_PatientID, PatientData, CT):-
                    CT='nct00002720',
                    currentlyPregnant(PatientData).
```

We formalize the criteria which have not been checked in a rule like this:

```
getNotYetCheckedItems(CT, NotYetCheckedItems):-
                    CT='nct00002720',
                    Item1 = 'No serious disease that would preclude surgery',
                    Item2 = 'No other prior or concurrent malignancy except
                      basal cell carcinoma or carcinoma in situ of the cervix',
                    NotYetCheckedItems = [Item1, Item2].
```

4.2 Formalization of Domain-specific Knowledge

We consider patient data as a set of property-value pairs. A general format of patient data, called the PrologCMR format, is designed to be a list of property-value pairs, like this:

```
[gender:Gender,
 birthyear:BirthYear,
 menopause:Menopause,
 currentlyPregnant:Pregnant,
 currentlyNursing:Nursing,
 diagnosis:Diagnosis,
 diagnosisyear:DiagnosisYear,
 her2:HER2,
 er:ER,
 pr:PR,
 stage:Stage]
```

The values in the pairs of the Prolog CMR format can be a term (i.e., a string or a number) or a list with the PrologCMR format. Namely it allows for a tree-structured data. For example, we can merge the properties of hormone receptors in the list above into a property-value pair, like this: hermone_receptor_status:[her2:HER2, er:ER, pr:PR].

This general format of patient data is flexible enough to represent the data from different formats of CMRs, because we can design a CMR-specific interface to obtain the corresponding data via different data servers, which can be a SPARQL endpoint, internal data storage server, or a database server. Then, we can convert the patient data into the PrologCMR format. We introduce the general predicate getItem(PatientData, Property, Value) to get the value of the property from the patient data.

Several receptor status of breast cancer cells have been considered to be very important for the classification of breast cancer. Those important receptors are estrogen receptor (ER), progesterone receptor (PR), and Human Epidermal growth factor Receptor 2(HER2). These receptor status can be straight forward formalized as follows:

[6] identifier from clinicaltrial.gov

```
er_positive(PatientData):- getItem(PatientData, er, ER), ER = 'positive'.
er_negative(PatientData):- getItem(PatientData, er, ER), ER = 'negative'.
```

Similarly we can define the predicates for PR and HER2.

More complex criteria where real domain knowledge is involved for instance the triple-negative breast cancer status which means that a patient of breast cancer is triple negative if she is ER negative, PR negative, and HER2 negative. This can be formalised as follows:

```
triple_negative(PatientData):-
                      er_negative(PatientData),
                      pr_negative(PatientData),
                      her2_negative(PatientData).
```

The menopausal status of a female patient is simply considered as a value of a property in the patient data. Actually in medical science, menopausal status is defined in terms of menstrual periods.

```
last_time(Patient, menstrual_period, LastMenstrucalPeriod):-
hasPatientData(Patient,PatientData),
      postmenopausal(PatientData),
      today(Today),
      at_least_earlier(LastMenstrucalPeriod, Today, 1, year).
```

The definitions of those temporal predicates (e.g. *at_least_earlier*) belong to the common knowledge level.

4.3 Formalization of Common Knowledge

Temporal Reasoning The rules for formalizing temporal reasoning and others are not domain-specific, because that kind of knowledge can be used in different applications. Thus, they are designed to be separated libraries, which are different from the domain specific libraries.

For reasoning with breast cancer knowledge, we may need various temporal operators(i.e., predicates), like those "before", "after", "until", "today", "no less than 6 months before", etc. Such general temporal operators are well known from the AI literature [4].

To summerise the general framework has three knowledge types: (1) clinical trial specific knowledge, (2) domain-specific knowledge, and (3) common knowledge. The clinical trial specific knowledge specified the eligibility criteria in terms of predicates defined in the domain-specific level if they are domain dependent and in the common knowledge level if they are domain independent but common knowledge. The levels (2) and (3) are the reusable parts for the formalisation of eligibility criteria whereas (1) use those re-usable parts in the formalisation of a specific eligibility criteria.

5 Implementation and Feasibility Experiment

5.1 Implementation

SWI-Prolog provides a powerful Semantic Web library, by which we can achieve semantic interoperability in the rule-based formulation of eligibility criteria efficiently and effectively. SWI-Prolog handles the semantic web RDF model and OWL data naturally. RDF and OWL provide stable models for knowledge representation with nice support for semantic interoperability.

The rule-based formulation of eligibility criteria of clinical trials is developed with the support of the following two semantic web libraries in SWI-Prolog:

Web-server and client library. This is the core semantic web package of SWI-Prolog. It provides an HTTP server and client, session handling, authorization, logging, etc, and libraries for generating HTML pages and JSON. Based on this library, we developed the interface with SPARQL endpoints to obtain semantic data for the rule-based formulation of eligibility criteria. (e.g. patient data and medical ontologies)

For example, the following rule in Prolog is designed to obtain the patient data for a SPARQL endpoint, which is located at the localhost with the port '8183':

```
getPatientData(PatientID, PatientData):-
      get_sparql_query(patientdata, Query, PatientID),
      findall(Row, (sparql_query(Query, Row,[host('localhost'), port(8183),
                  path('/sparql/')])), Answers),
      sparql_answer_to_list(patientdata(PatientID), Answers, PatientData).
```

Namely: given a patienID, the predicate 'getPatientData' would return the patient data from the corresponding SPARQL endpoint. We use the predicate get_sparql_query(patient, Query, PatientID) to get a system specific SPARQL query for the given patient ID. We use the built-in predicate sparql_query to obtain the result Answers from the SPARQL endpoint. We design a predicate sparql_answer_to_list to convert the answers from the SPARQL endpoint into the internal representation of the patient data (i.e., a Prolog list), so that the patient data can be processed further by the predicate getItem, as we have discussed in the section about the formalization of domain specific knowledge.

RDF storage and query library. SWI-Prolog provides powerful support for the storage and manipulation of semantic data, like loading and saving RDF data and querying them. This RDF library loads and saves XML/RDF and Turtle. It also provides simple RDFS and OWL support which is sufficient for the temporary internal storage of semantic data in the rule-based formulation of eligibility criteria.

5.2 Feasibility

SemanticCT[7] is a semantically enabled system for clinical trials. The goals of SemanticCT are not only to achieve interoperability by semantic integration of heterogeneous data in clinical trials, but also to facilitate automatic reasoning and data processing services for decision support systems in various settings of clinical trials.

SemanticCT is built on the top of the LarKC (Large Knowledge Collider) platform[8], a platform for scalable semantic data processing. We have implemented the rule-based formulization of eligibility criteria as a component of SemanticCT for the service of automatic identification of eligible patients for clinical trials.

Experiment design: An ideal experimental scenario would look as follows: (i) take realistic corpus of patient records for included and excluded patients for a given set of trials; (ii) formalise the inclusion and exclusion conditions for these trials; (iii) execute these formalised criteria on the data of included and excluded patients; and (iv) compare precision and recall of the automatically selected set of patients against the actual selections as given in the corpus.

Available data: A corpus of current and past clinical trials is readily available [9] but given the lack of a realistic collection of patient data for such trials, we limit ourselves in this paper to a *feasibility* study that shows how two important tasks can in principle be supported by the formalisation and implementation that we discussed above. Our experiments concern a *patient recruitment* task (= finding patients that qualify for a given trial), and a *trial feasibility* task (=

[7] http://wasp.cs.vu.nl/sct
[8] http://www.larkc.eu
[9] e.g. clinicaltrial.gov

checking if a set of inclusion and exclusion criteria for a newly designed trial results in a sufficient number of recruitable patients).

For a small corpus of clinical trials in our experiments, we have picked up 10 clinicial trials of breast cancer out of the 4665 NCT clinical trials (1,200,565 triples) and we formalized the eligibility criteria of those selected trials (the trial ID numbers are listed in the tables that follow).

For patient data, we generated a set of 10,000 plausible patient files created by our Knowledge-based Patient Data Generator[10]. This Knowledge-Based Patient Data Generator uses clinical and epidemiological background knowledge to generate a patient population that is both medically plausible, and that has a realistic statistical distribution.

Experiment 1: Patient Recruitment: Some eligibility criteria cannot be checked automatically over the patient data, because they need additional input from patients, like the criteria 'Patients must be mentally competent to understand and give informed consent for the protocol'. Some eligibility criteria have not yet been checked, because their corresponding data have not yet been available in the existing patient data format, like the criteria 'Must have regular menstrual cycles (21-35 days)'.

Table 1 reports on a (simulated) *patient recruitment scenario*, and summarises how many criteria have been checked in the test. The table shows that we can check maximally 83.33% of the criteria, and minimally 34.48% of the criteria, based on the given patient data.

Clinical Trial ID	Total Criteria	Checked Criteria	Total IC	Total EC	Checked IC	Checked EC	NYC IC	NYC EC	Checked Criteria Rate(%)
NCT00001250	22	15	11	11	7	8	4	3	68.18
NCT00001385	18	15	9	9	7	8	2	1	83.33
NCT00002720	10	7	9	1	7	0	2	1	70.00
NCT00002762	15	7	10	5	5	2	5	3	46.67
NCT00002934	26	10	20	6	9	1	11	5	38.46
NCT00003329	6	3	5	1	3	0	2	1	50.00
NCT00003654	18	11	1 0	8	9	2	1	6	61.11
NCT00005023	29	10	27	2	9	1	18	1	34.48
NCT00005079	18	11	10	8	7	4	3	4	61.11
NCT00005587	16	10	12	4	10	0	2	4	62.50

Table 1. Checked Eligibility Criteria. IC:Inclusion Criteria, EC: Exclusion Criteria, NYC: Not Yet Checked Items

Experiment 2: Trial Feasibility In this feasibility experiment, we use our system to automatically determine if a given target number of patients can be recruited for a trial: T200 stands for finding 200 candidate patients, and T500 stands for finding 500 candidate patients, from the total 10,000 patients. Table 2 shows the results of trial feasibility with different targets. For the lower target (e.g., T200), we can always find the targeted numbers of the candidate patients who meet the checked criteria. For the higher target (e.g., T750), we cannot find enough candidate patients for four trials. Furthermore, a lower percentage of checked items does not necessary lead to higher recruitment rate. For example, Trial 'NCT00002762' (with checked item rate 46.67) can find only 32.13 percent of the target. That means that some of checked criteria in this trial have low feasibility, and the limited number of the patient data also lead to the difficulties.

These experiments show that conditions of realistic trials can be formalised and implemented in such a way that, at least on our artificially generated but medically and statistically plausible patient data, both patient recruitment and trial feasibility can be supported.

[10] http://wasp.cs.vu.nl/apdg

Clinical TrialID	T200 Founded	T200 Rate(%)	T300 Founded	T300 Rate(%)	T500 Founded	T500 Rate(%)	T750 Founded	T750 Rate (%)
NCT00001250	200	100	300	100	500	100.00	750	100
NCT00001385	200	100	300	100	500	100	750	100
NCT00002720	200	100	300	100	397	79.40	397	52.93
NCT00002762	200	100	241	80.33	241	48.20	241	32.13
NCT00002934	200	100	300	100	500	100	750	100
NCT00003329	200	100	300	100	500	100	750	100
NCT00003654	200	100	300	100	500	100	750	100
NCT00005023	200	100	300	100	500	100	501	66.80
NCT00005079	200	100	281	93.67	281	56.20	281	37.47
NCT00005587	200	100	300	100	500	100	750	100

Table 2. Trial Feasibility

6 Related Work, Discussion and Conclusion

6.1 Related Work

An extensive survey of formal representations of eligibility criteria appears in [5]. Below we discuss a subset of papers that can be directly compared to our own work.

In [6] the authors translate each free-text eligibility criterion into a machine executable statement using a derivation of the Arden Syntax. Clinical trial protocols were then structured as collections of these eligibility criteria using XML. In our work, we use a more expressive rule-based language and then structured the eligibility criteria as RDF.

[7] presents a method entirely based on standard semantic web technologies and tools, that allows the automatic recruitment of a patient to available clinical trials. They use a domain specific ontology to represent data from patients' health records and use SWRL to verify the eligibility of patients to clinical trials. Although we propose an even more expressive language for modelling the eligibility criteria this is in the same spirit as our approach. Furthermore, we proposed a general framework for specifying the eligibility criteria in three types of knowledge to facilitate reuse.

The purpose of [8] is to develop algorithms that automatically identify qualified patients for breast cancer clinical trials from free-text medical reports. Similarly, TrialX [9] is a consumer-centric tool that matches patients to clinical trials by extracting their PHR information and linking it to the most relevant clinical trials using semantic web technologies. As a further example, [10] annotates free text criteria with ERGO annotations. The matching algorithms in these works combine semantic and NLP techniques. In our approach we first do the modeling of the eligibility criteria based on the three different levels of knowledge in a rule-based approach and then we do the matching against the patient data automatically.

The empirical analysis in [11] shows that the vast majority (85%) of trial criteria is of "significant semantic complexity". This justifis our choice for an expressive rule-based formalism. The paper also observes that temporal data play a role in 40% of all criteria, justifying our choice for a separate layer for this in our formalisation.

Our rules are currently limited to boolean yes/no judgements on criteria. The work in [12] also considers a probabilistic approach. This would be an interesting future extension of our formalism.

6.2 Discussion and Conclusion

For clinical trials, identifying eligible patients are mostly manually. Thus, it often results in low clinical trial enrolment. The formulation of eligibility criteria provides the possibility for faster

identification of patients for clinical trials. Based on the rule-based formulation of eligibility criteria, we can achieve an efficient way for automatic identification of eligible patients whenever possible.

With the support of the Semantic Web library in the Prolog-based formalism, we can achieve the semantic interoperability among EHR and clinical trial systems, because the relevant can be exploited to allow more efficient patient enrolment in clinical trials. Semantic interoperability between EHR and CT systems enables us to provide solutions for patient recruitment that help avoid double data entry: establishing a single source for each data item, automatic storing of clinical trial eligibility criteria into the EHR and using the EHR data for automatic Electronic Data Capture (EDC).

However, automatic patient recruitment for clinical trials would not be considered as a simple system of automatic checking with the criteria of patient data. In many application scenarios of patient recruitment, it should be considered to be one with a decision making system, which involves complex procedure and comprehensive processing over various data and workflows. Furthermore, it would be also beneficial if the formalism can accommodate and integrate with the clinical guidelines for specific diseases [13]. That requires that rule-based formulation of eligibility criteria can be extended with more powerful workflow processing. We will leave the integration of rule-based formulation of eligibility criteria with integrated guidelines as one of the future work.

Acknowledgment This work is partially supported by the European Commission under the 7th framework programme EURECA Project.

References

[1] Bucur, A., ten Teije, A., van Harmelen, F., Tagni, G., Kondylakis, H., van Leeuwen, J., Schepper, K.D., Huang, Z.: Formalization of eligibility conditions of ct and a patient recruitment method, deliverable d6.1. Technical report, EURECA Project (2012)

[2] Huang, Z., ten Teije, A., van Harmelen, F.: SemanticCT: A semantically enabled system for clinical trials. Technical report, under preparation (2012)

[3] Wielemaker, J., Schrijvers, T., Triska, M., Lager, T.: Swi-prolog. Journal of Theory and Practice of Logic Programming (1-2) (2012) 67–96

[4] Allen, J.F.: Maintaining knowledge about temporal intervals. CACM **26**(11) (1983) 832–843

[5] Domschke, S., Domschke, W.: Antirheumatic drug-induced damage of the gastroduodenal mucosa: approach to prevention. Internist (Berl) **27**(10) (Oct 1986) 630–636

[6] Ohno-Machado, L., Wang, S.J., Mar, P., Boxwala, A.A.: Decision support for clinical trial eligibility determination in breast cancer. Proc AMIA Symp (1999) 340–344

[7] Besana, P., Cuggia, M., Zekri, O., Bourde, A., Burgun, A.: Using semantic web technologies for clinical trial recruitment. In: International Semantic Web Conference. (2010) 34–49

[8] Zhang, J., Gu, Y., Liu, W., Hu, W., Zhao, T., Mu, X., Marx, J., Frost, F., Tjoe, J.: Automatic patient search for breast cancer clinical trials using free-text medical reports. In: ACM International Health Informatics Symposium, IHI 2010, ACM (2010) 405–409

[9] Patel, C., Gomadam, K., Khan, S., Garg, V.: Trialx: Using semantic technologies to match patients to relevant clinical trials based on their personal health records. J. Web Sem. **8**(4) (2010) 342–347

[10] Tu, S.W., Peleg, M., Carini, S., Bobak, M., Ross, J., Rubin, D., Sim, I.: A practical method for transforming free-text eligibility criteria into computable criteria. J Biomed Inform **44**(2) (Apr 2011) 239–250

[11] Ross, J., Tu, S., Carini, S., Sim, I.: Analysis of eligibility criteria complexity in clinical trials. AMIA Summits Transl Sci Proc **2010** (2010) 46–50

[12] Tu, S.W., Kemper, C.A., Lane, N.M., Carlson, R.W., Musen, M.A.: A methodology for determining patients' eligibility for clinical trials. Methods Inf Med **32**(4) (Aug 1993) 317–325

[13] de Clercq, P., Blom, J., Korsten, H., Hasman, A.: Approaches for creating computer-interpretable guidelines that facilitate decision support. Artif Intell Med (1) (May 2004) 1–27